REMARKS

Applicants have carefully considered this Application in connection with the Examiner's Final Office Action of December 28, 2007, and respectfully request reconsideration of this Application in view of the above amendment and the following remarks.

Pending in this application are Claims 1, 6-13, 19-22, 24, 26, 28, 58, 61-63, 65-68, 71 and 72.

Applicants have amended Claim 21 to provide that the coordinate system is a three-dimensional coordinate system. This is consistent with the specification and the claims, particularly Claim 11.

I. Rejections Under 35 U.S.C. §103(a)

A. <u>Claims 1, 6, 7, 11-13, 19-22, 24 , 58, 61-63, 65-68, and 71</u>

Claims 1, 6, 7, 11-13, 19-22, 24 , 58, 61-63, 65-68, and 71 stand rejected under 35 U.S.C. §103(a) as being unpatentable over the NPL Document "Automated Recognition of Intersecting Features from 2-D CAD for Collaborative Virtual Prototyping" by Ganesan ("Ganesan"), in view of U.S. Patent No. 7,149,677 to Jayaram et al. ("Jayaram"). The Examiner asserts that Ganesan teaches nearly all of the elements of the claims. The Examiner admits that Ganesan does not teach the step of storing the representation in an intermediate binary file format, but asserts that Jayaram teaches this limitation. Applicants respectfully assert that Ganesan in combination with Jayaram does not teach every limitation of these claims and would respectfully draw the Examiner's attention to the reasoning below.

1. Claim 1

Despite the Examiner's assertions otherwise, Ganesan combined with Jayaram does not teach all of the elements of Claim 1. In particular, Ganesan combined with Jayaram fails to teach or suggest:

- Ordering the plurality of built features..., wherein the plurality of built features are **ordered consistently with the CAD system's ordering expectations**

Ganesan does not teach or suggest the limitation of ordering the plurality of built features consistently with the CAD system's ordering expectations. The Examiner asserts that this is shown in Ganesan's Figure 3.1 and is discussed on page 47 where Ganesan states "the features that have now been identified can be assembled to recreate the feature model of the input design." Applicants assert that this does not demonstrate ordering of any kind, and particularly does not demonstrate ordering consistently with a CAD system's ordering expectations. CAD systems use complicated ordering systems called dependence rules, which involve the determination of base features followed by the dependence of child features. Ganesan's references to hierarchies and to assembling features do not constitute teachings of this kind of ordering, but are rather merely references to Ganesan's post-processing and verification methods. No particular order is required by Ganesan.

- Building a 3-D feature-based model **based on the ordering of the plurality of built features** to give a representation

As already discussed, Ganesan does not teach or suggest ordering the plurality of built features in any way, much less according to the CAD system's ordering expectations, as required by Claim 1. Thus, Ganesan also does not teach building a 3-D feature-based model based on the ordering of the plurality of built features. The Examiner's references to page 47 of Ganesan are again unconvincing, as page 47 merely mentions the creation of the feature model from the identified features thus far. There is simply no reference or mention of determining and/or using ordering of these features according to CAD-specific ordering expectations in the model creation in Ganesan. The ordering that is required by Claim 1 cannot simply be implied from Ganesan's mentions of model creation as this is an important distinction and must be specifically taught.

- Storing the **representation** in an intermediate binary file format (Claim 1).

The Examiner admits that Ganesan does not teach this limitation. Applicants now assert that Jayaram also does not teach storing the representation in an intermediate binary file format. The Examiner cites to excerpts of Jayaram that talk about extracted data being stored in a designated metafile format, referred to as an intermediate file format. Applicants assert that these portions of

Jayaram cited by the Examiner do not refer to storing a representation, as required by Claim 1. Rather, these portions of Jayaram merely refer to storing extracted point cloud data. This is clear from simply reading this sentence in context with its immediately preceding sentence, which state "Furthermore, comparison reference data (or point cloud data) is then extracted. Finally, the extracted data is stored in a designated metafile format (an intermediate file format)." See Jayaram, Col. 16, ll. 38-41. The intermediate storage of this point cloud data has nothing to do with storing the representation in an intermediate binary file format. The term representation in Claim 1 refers to the 3-D feature-based model data, not point cloud data, which is clear from the language of the claims where the words "geometric representation" are used.

For these reasons, Ganesan in combination with Jayaram does not teach or suggest all of the limitations of Claim 1.

2. Claim 6

With regard to Claim 6, Ganesan combined with Jayaram also fails to teach or suggest:

- Wherein **the intermediate binary file format comprises** a geometry library and a feature library adapted to build the three-dimensional model

As already discussed with regard to Claim 1, upon which Claim 6 depends, Jayaram does not teach or suggest the intermediate binary file format. In this rejection of Claim 6, the examiner has attempted to reference unrelated terms in Jayaram's Table 2 and Col. 14, ll. 28-35. Firstly, it is the intermediate binary file format in Claim 6 that contains the feature and geometry library. As shown previously, Jayaram's metafile format stores the point cloud data. Secondly, Table 2 merely illustrates a specific case (See Col. 13, ll. 27-32) mapping one particular feature type (hole) between two CAD systems viz Pro/Engineer and CATIA. The examiner misinterprets the discussion in Col. 14, ll. 28-35. Here, Jayaram discusses the complexity of the geometry with reference to the performance of the mapping algorithms. This has no relation to Claim 6, which refers to the feature and geometry library contained in the intermediate binary file format. For this reason, Jayaram also does not teach or suggest the additional limitations of Claim 6.

3. Claim 7

With regard to Claim 7, Ganesan combined with Jayaram also fails to teach or suggest:

- Wherein the geometry library comprises geometry classes for: two-dimensional entities; three-dimensional entities; arc; **elliptical arc; polyline;** spline; **face; points; and vectors**

In addition to the underlying limitations of Claim 1, which Claim 7 depends on, Ganesan in combination with Jayaram also does not teach the limitations of Claim 7. Claim 7 requires a geometry library including 3-D entities. Ganesan operates exclusively in the 2-D space. This is clear from many statements made in Ganesan, including the statement at page 129 regarding the advantages of the method that reads "these algorithms operate in 2-D space," as well as the statement at page 42 which reads "[t]he core of the system is a unique feature extraction system in the 2-D domain." The citation of the Examiner to Figures 3.2 and 3.3 of Ganesan are unconvincing. Figures 3.2 and 3.3 describe the parameters defining the isolated features. These parameters are in 2-D space only. In all of Ganesan, only simple 2D entities (line and arc) are considered. Claim 7 requires other types such as elliptical arc and polylines. Ganesan's architecture does not store any 3-D data such as face/plane/coordinate system information defined by a 3-D point and vector. The end result of the Flow diagram in Figure 3.1 states "3-D Feature-based CAD" but there is no mention of 3-D geometry data being stored. Thus, Ganesan does not teach a geometry library having these 3-D entities and rather teaches away from the storage of these entities due to its emphasis on 2-D entities. For these reasons, Ganesan in combination with Jayaram does not teach or suggest all of the limitations of Claim 7.

4. Claim 11

With regard to Claim 11, Ganesan combined with Jayaram also fails to teach or suggest:

- Independently defining each feature via a **three-dimensional coordinate system**

In addition to the underlying limitations of Claim 1, which Claim 11 depends on, Ganesan in combination with Jayaram also does not teach the limitations of Claim 11. Claim 11 requires a geometry library including 3-D entities. As discussed above, Ganesan operates exclusively in the 2-D space. This is clear from the statements made in Ganesan already addressed above, including the statement at page 129 regarding the advantages of the method that reads "these algorithms operate in 2-D space," as well as the statement at page 42 which reads "[t]he core of the system is a unique feature extraction system in the 2-D domain." Adding to the point made in the previous argument, the features presented in Ganesan's architecture are limited only to parametric data as shown in Figures 3.2 and 3.3 and Appendix data which are exclusively in 2-D form. Claim 11 requires a 3-D coordinate system based on which all the profile-based features are drawn in the claimed 3-D CAD system. For these reasons, Ganesan in combination with Jayaram does not teach or suggest all of the limitations of Claim 11.

4. Claim 12

With regard to Claim 12, Ganesan combined with Jayaram also fails to teach or suggest:

- Wherein **the three-dimensional coordinate system** contains the data necessary to detect at least one of a following element from a group consisting of: **a work plane; a sketch plane; and a face** upon which a feature may need to be built

In addition to the underlying limitations of Claim 1, which Claim 12 depends on, Ganesan in combination with Jayaram also does not teach the limitations of Claim 12. Again, this claim requires a three-dimensional coordinate system that Ganesan simply does not teach. In general CAD parlance, work/sketch planes and existing faces are 3-D entities required to correctly represent profile-based features. Claim 12 refers to the components of the 3-D coordinate system object used in definition such planes. The sections cited by the examiner in Ganesan do not refer to identifying the 3-D details of plane/face. The discussion on page 48 mentions that Figure 3.2 only illustrates the isolated features with the parametric attributes. Similarly, on page 57 it is mentioned that Figure 3.6 shows the representation of the square slot, as projected on to the orthographic views. Nowhere in Ganesan's document is the detection of 3-D work/sketch plane or existing face entities discussed.

For these reasons, Ganesan in combination with Jayaram does not teach or suggest all of the limitations of Claim 12.

5. Claim 13

With regard to Claim 13, Ganesan combined with Jayaram also fails to teach or suggest:

- Wherein the data comprises at least one of a following element from a group consisting of: **plane vectors; an origin of the plane; and an elevation of the plane** from a world origin

In addition to the underlying limitations of Claim 1, which Claim 13 depends on, Ganesan in combination with Jayaram also does not teach the limitations of Claim 13. As discussed with regard to Claim 12, since Ganesan does not teach the detection of the plane/face to build a 3-D feature, the definition elements of 3-D plane vectors or 3-D origin and elevation distance are never mentioned nor determined in Ganesan. Again, Ganesan is focused entirely on 2-D space. For these reasons, Ganesan in combination with Jayaram does not teach or suggest all of the limitations of Claim 13.

6. Claim 19

With regard to Claim 19, Ganesan combined with Jayaram also fails to teach or suggest:

- Wherein the feature constraints are handled **via a class that provides at least one of a following action from a group consisting of…**

In addition to the underlying limitations of Claim 1, which Claim 19 depends on, Ganesan in combination with Jayaram also does not teach the limitations of Claim 19. Ganesan uses the concept of constraints to **only detect** the 2-D shape of a certain feature like a square slot (see Ganesan, Figure 3.4.1). This is not related to the constraint that is **stored** in Claim 19 to correctly position a feature in 3-D model space. For these reasons, Ganesan in combination with Jayaram does not teach or suggest all of the limitations of Claim 19.

Claim 20 depends from Claim 1 and therefore, for the reasons discussed above, is also not obvious in view of Ganesan combined with Jayaram.

7. Claim 21

With regard to Claim 21, Ganesan combined with Jayaram also fails to teach or suggest:

- Wherein each **view class** contains at least one of a following element from a group consisting of: an array of two-dimensional entities; and **a <u>three-dimensional</u> coordinate system** associated with the view

In addition to the underlying limitations of Claim 1, which Claim 21 depends on, Ganesan in combination with Jayaram also does not teach the limitations of Claim 21. Claim 21 has been amended to provide that the coordinate system is a three-dimensional coordinate system. As already discussed, Ganesan's architecture does not store the 2-D view configuration in any intermediate format. Thus, no 3-D coordinate system detail is stored in the approach. Claim 21 offers the possibility of propagating the input view configuration via this view class for subsequent feature validation. This is simply not taught or suggested by Ganesan. For these reasons, Ganesan in combination with Jayaram does not teach or suggest all of the limitations of Claim 21.

8. Claims 22, 24

With regard to Claims 22 and 24, Ganesan combined with Jayaram also fails to teach or suggest:

- Transferring system/application specific data through an intermediate binary file **based on the ordering of the built features**

In addition to the underlying limitations of Claim 1, which Claims 22 and 24 depend on, Ganesan in combination with Jayaram also does not teach the limitations of Claims 22 and 24. While it is true that these claims and Ganesan tend to use the similar terms "system/application specific data," this does not indicate that Ganesan teaches these limitations as they are defined in the claims. Furthermore, Jayaram's approach does not explicitly store ordering information of the features. As stated in Jayaram at Col. 14, ll. 46-50 and Box 90 of Figure 2, the conversion process is at the underlying native-geometry level and not at the higher feature-level. For these reasons, Ganesan in combination with Jayaram does not teach or suggest all of the limitations of Claims 22 and 24.

9. Claim 58

Despite the Examiner's assertions otherwise, Ganesan combined with Jayaram does not teach all of the elements of Claim 58. In particular, Ganesan combined with Jayaram fails to teach or suggest:

- Using an automated feature detection system **to create matched feature loops**

The discussion of the Feature Extraction system in Ganesan mentions the following in bottom of Page 45. *"The feature extraction system does not directly extract features, but extracts feature subparts"*. Feature subparts are not the same as the matched feature loops of Claim 58. Matched feature loops correspond to the representation of the potential 3-D feature if projected back to the orthographic planes.

- Performing a **profile analysis** and a feature analysis on the matched feature loops

Profile analysis refers to the analysis of the matched feature loops required to evaluate and select the feature loop that would be the most pertinent candidate for the profile. Ganesan's architecture does not consider a profile analysis step in its feature identification algorithms.

- Producing an **ordered list of three-dimensional features** using geometry of **up to six orthographic views** of the three-dimensional features, **wherein the ordered list of three-dimensional features is ordered consistently with the CAD system's ordering expectations**

As discussed above, the concept and step of ordering of the individual features has not been considered in Ganesan's architecture. Further, use of the geometry of up to six orthographic views (any combination of front, back, top, bottom, right or left) in the input 2-D drawing can be considered to build the 3-D model. In his discussions, Ganesan considers a three-view configuration only, such as in sections 3.2.1 and 3.2.2 in Page 53. At page 132, Ganesan states that his algorithms are limited to "three orthographic views." Thus, Ganesan clearly does not teach using up to six orthographic views.

- Writing the **ordered list of three-dimensional features** to an intermediate binary file format

As discussed previously in earlier arguments for Claim 1, Jayaram's step referenced by the examiner (Col.16, ll. 40-41) deals with writing of the point cloud data. This is not related to the step of writing the ordered list of 3-D features in Claim 58.

- **Interfacing the binary file format to a CAD system-specific binary file format**

This step refers to the mapping of the intermediate binary file format (containing 3-D feature data) to the file format of the target 3-D CAD system. As already discussed, Jayaram's intermediate file format stores the point cloud data from the source model and not the list of 3-D features and related data.

For these reasons, Ganesan in combination with Jayaram does not teach or suggest all of the limitations of Claim 58. Claims 61-63, which are dependent upon Claim 58, are also not obvious in view of Ganesan combined with Jayaram.

10. Claim 65

With regard to Claim 65, Ganesan combined with Jayaram also fails to teach or suggest:

- **Splitting entities** in the drawing or in the corrected drawing corresponding to top, front and side views

The Examiner cites to Figure 3.6 of Ganesan as showing the step of automatically splitting entities, but provides no further explanation of this citation. Applicants respectfully assert that Figure 3.6 does not illustrate what the Examiner appears to be claiming that it does. Figure 3.6 as cited by the Examiner merely illustrates the views for a square slot. The process of splitting drawing entities as part of identifying the orthographic view configuration is not discussed in Ganesan and is certainly not shown in Figure 3.6.

- **Performing error checking** on the drawing and if errors are found, correcting the errors

Applicants admit that the term "filtering non-graphical entities" found in Claim 65 can also be found in Hazama. However, the use of similar terminology is not relevant when the terminology is used in reference to two different processes. The clean-up process of bendlines and trim faces

discussed in Hazama (Col. 58, ll. 20-24) that is cited by the examiner occurs at the 3-D part stage. Claim 65 clearly refers to the correction of the input **2-D drawing** elements. This is clear because steps (a) and (b) of Claim 65 clearly refer back to steps (a) and (b) of Claim 58, on which Claim 65 depends. Steps (a) and (b) of Claim 58 clearly refer to "the two-dimensional drawing." Thus, these teachings of Hazama are not applicable.

- **Exploding any blocks** in the drawing to accumulate indivisible geometric entities

Hazama's teachings of detecting inside loops, holes, and shapes are also not relevant to the claimed step of exploding any blocks in the drawing. Claim 65 refers to the process of breaking any blocks in a drawing to create multiple, indivisible, geometric entities. This is not related to the purpose of Figure 14A, Step 192 of Hazama. The description in Col. 55, ll. 36-52 of Hazama points to the process of determining closed inside loops in a 2-D view containing the profile for the sheetmetal cutout. This is not the same step that is being accomplished in Claim 65.

In addition to the underlying limitations of Claim 58, which Claim 65 depends on, Ganesan in combination with Jayaram also does not teach the limitations of Claim 65. For these reasons, Ganesan in combination with Jayaram does not teach or suggest all of the limitations of Claim 65.

11. Claims 66-68

With regard to Claim 13, Ganesan combined with Jayaram also fails to teach or suggest:

- Fixing a common origin for each view...**translating the entities to the common origin...writing translated geometric entity data to classes**

In addition to the underlying limitations of Claim 58, which Claims 66-68 depend on, Ganesan in combination with Jayaram also does not teach the limitations of Claims 66-68. Ganesan does not fix a common origin as is required in these claims. The Examiner suggests that Ganesan's description within section 3.2.1 that states "the maximum values of x and y coordinates of the entities" can establish a common origin for each view. However, the above sentence is in context of creating of bounding box for the view, Ganesan does not consider translation of geometric entities to the common origin or the writing of this translated data to classes anywhere in his architecture. As is

described in the specification, the lower left corner of every view of the current invention is fixed to be the common origin and the end points of all geometric entities of the view are recalculated based on this common origin. Ganesan clearly does no such fixing. For these reasons, Ganesan in combination with Jayaram does not teach or suggest all of the limitations of Claims 66-68.

Claim 71, which is dependent upon Claim 58, is also not obvious in view of Ganesan combined with Jayaram.

B. <u>Claims 8-10</u>

Claims 8-10 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Ganesan, in view of Jayaram, and in further view of the NPL Document "C++ Data Structures" by Dale ("Dale"). The Examiner asserts that Ganesan in combination with Jayaram teaches nearly all of the elements of the claims. The Examiner admits that Ganesan does not teach the step of storing the representation in an intermediate binary file format, but asserts that Jayaram teaches this limitation. Finally, the Examiner argues that Dale teaches the remaining limitations of Claims 8-10.

1. Claims 8-9

With regard to Claims 8-9, Ganesan combined with Jayaram and Dale fails to teach or suggest:

- **Copying data** between at least one of the class's private data space and an address of the data...further comprising, within each class, classifying the data as at least one of...fundamental data; and derived data, wherein derived data is any **additional information** that may be used during the building of the 3-D model (Claims 8-9)

In addition to the underlying limitations of Claim 1, which Claims 8-9 depend on, Ganesan in combination with Jayaram and Dale also does not teach the limitations of Claims 8-9. In particular, Dale does not teach the step of copying data as this step is defined in the claims. The Copy function referenced by the Examiner in Dale (pages 344-45) deals with the capabilities of a generic friend function. The base/derived class parlance cited by the examiner in Dale (Page 363) refers to the working of objected inheritance concept. Claim 8 refers to a generic copy function, with no extra

qualifiers added. This claim refers to steps required to support the transfer of 2-D/3-D geometric object data.

Claim 9 refers to highlighting the difference between fundamental (i.e., necessary for completely defining a 2-D geometric entity) and derived (i.e., auxiliary data which can be calculated from fundamental information and which by itself is useful in performing some geometric analysis) member data. They both reside in the same class definition and henceforth have no relevance to the concept of the object inheritance in C++ as is being argued by the Examiner. A trivial example to highlight the difference is for a 2-D circle object. The fundamental data includes center point and radius whereas the derived data could include perimeter, area, etc. This is simply not taught or suggested by Dale, which refers to object inheritance in C++ alone. For these reasons, Ganesan in combination with Jayaram and Dale does not teach or suggest all of the limitations of Claims 8-9.

2. Claim 10

With regard to Claim 10, Ganesan combined with Jayaram also fails to teach or suggest:

- Ensuring, by each of the classes, that any change made to the fundamental data via a function **will update the derived data accordingly**

In addition to the underlying limitations of Claim 1, which Claim 10 depends on, Ganesan in combination with Jayaram and Dale also does not teach the limitations of Claim 10. As discussed with regard to Claims 8-9, Dale simply does not teach the existence of fundamental data and derived data as those terms are used in the claimed subject matter. Dale is limited to C++ object inheritance and does not discuss the transfer of 2-D/3-D geometric object data. In the current claim, any change to the fundamental member data of a 2-D geometry object will lead to the update of the dependent derived member data in the same class. This is unique to the claimed subject matter and is not taught by Dale because it is simply not part of Dale's C++ object inheritance. For these reasons, Ganesan in combination with Jayaram and Dale does not teach or suggest all of the limitations of Claim 10.

C. <u>Claims 26 and 28</u>

Claims 26 and 28 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Ganesan, in view of Jayaram, and in further view of U.S. Patent No. 5,970,490 to Morgenstern ("Morgenstern"). The Examiner asserts that Ganesan in combination with Jayaram teaches nearly all of the elements of the claims. The Examiner admits that Ganesan does not teach the step of storing the representation in an intermediate binary file format, but asserts that Jayaram teaches this limitation. Finally, the Examiner argues that Morgenstern teaches the remaining limitations of Claims 26 and 28. Claim 26 depends on Claim 1 and thus because Ganesan in view of Jayaram fails to teach the limitations of Claim 1 as discussed above, Claim 26 is also not obvious in view of Ganesan, Jayaram, and Morgenstern. With regard to Claim 28, Ganesan combined with Jayaram and Morgenstern also fails to teach or suggest:

- Wherein the binary file format can be **incrementally** updated

The term that the Examiner appears to be ignoring in the claim is the term "incrementally." Claim 28 refers to the ability of updating incremental or only the most-recent changes made to the intermediate binary file. This is simply the ordinary meaning of the term incremental in the industry. The section of Morgenstern (Col. 29, ll. 36-39) referred to by the Examiner only discusses the sequential order of the reading of binary region and does not discuss anything about any variation in the reading process of a recently updated binary file. Thus there is no teaching of incrementally updating. For these reasons, Ganesan in combination with Jayaram and Morgenstern does not teach or suggest all of the limitations of Claim 28.

D. Claim 72

Claims 8-10 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Ganesan, in view of Jayaram, and in further view of U.S. Patent No. 6,629,065 to Gadh ("Gadh"). The Examiner asserts that Ganesan in combination with Jayaram teaches nearly all of the elements of the claims. The Examiner admits that Ganesan does not teach the step of storing the representation in an intermediate binary file format, but asserts that Jayaram teaches this limitation. Finally, the Examiner argues that Gadh teaches the remaining limitations of Claim 72. Applicant respectfully disagrees, and asserts that the limitations of Claim 58, upon which Claim 72 depends, are not taught

or suggested by Ganesan in view of Jayaram. In addition, Ganesan combined with Jayaram and Gadh fails to teach or suggest:

- **Performing a profile analysis** on each loop match
- Building **feature subtrees**, wherein each of the feature subtrees contains necessary data to create a 3-D feature
- Building a **model tree based** on the feature relations...producing a **final feature tree** based on the model tree **to give the ordered list of three dimensional features**

Claim 72 refers back to Claim 58(d). In Claim 58(d), the matched feature loops are still in 2-D form and are not converted to parametric feature format until the profile analysis is completed. The Examiner misinterprets Figure 3.10 and section 3.4.5 on page 64 of Ganesan as being similar in scope to profile analysis. Rather, it is merely an illustration to show the orthographic views for a particular feature i.e., open pocket. As discussed in a prior argument, profile analysis is a step dealing with the process of finding the most likely candidate among the matched feature loops to form the basis of the feature.

Further, Applicants acknowledge that the teachings in Figures 18-21 of Gadh appear to be similar to the claims. However, Applicants respectfully point out the following differences. First, the figures in Gadh are instances of graph structures (see Col. 13, l. 60 – Col. 14, l. 9), whereas the data structure used in the claimed feature subtrees are of tree type, which are a special case of graph. Second, Gadh's teachings do not require complete ordering information to create the design intent graph. For instance, in Figure 20b, nodes b2 and b3 are interchangeable and would not alter the state of the graph. On the other hand, the feature tree built at the end of step (e) in Claim 72 has a unique ordering placed within its features. Third, Figure 21 of Gadh finally produces a graph representation of the model. It is not the 3-D model as is. On the other hand, step (e) produces the feature model as visible on the 3-D CAD system along with the final feature history tree. For these reasons, Ganesan in combination with Jayaram and Gadh does not teach or suggest all of the limitations of Claim 72.

II. Conclusion

Applicants respectfully submit that, in light of the foregoing amendments and comments, Claims 1, 6-13, 19-22, 24, 26, 28, 58, 61-63, 65-68, 71 and 72 are in condition for allowance. A Notice of Allowance is therefore requested.

If the Examiner has any other matters which pertain to this Application, the Examiner is encouraged to contact the undersigned to resolve these matters by Examiner's Amendment where possible.

Respectfully submitted,

T. L. Chwang

T. Ling Chwang
Registration No. 33,590
JACKSON WALKER L.L.P.
901 Main Street, Suite 6000
Dallas, Texas 75202
Tel: (214) 953-5959
Fax: (214) 661-6870

June 30, 2008
Date